Infrastructure Asset Management

Real-time traffic event detection using Twitter: A case study

--Manuscript Draft--

Manuscript Number:	IAsMa-D-17-00022
Full Title:	Real-time traffic event detection using Twitter: A case study
Article Type:	Themed Issue: Highway infrastructure
Corresponding Author:	Angelica Milagros Salas Jones, MSc University of Wolverhampton Faculty of Science and Engineering Wolverhampton, West Midlands UNITED KINGDOM
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	University of Wolverhampton Faculty of Science and Engineering
Corresponding Author's Secondary Institution:	
First Author:	Angelica Milagros Salas Jones, MSc
First Author Secondary Information:	
Order of Authors:	Angelica Milagros Salas Jones, MSc
	Panagiotis Georgakis, PhD
	Ioannis Petalas, PhD
	Renukappa Suresh, PhD
Order of Authors Secondary Information:	
Abstract:	Incident detection is an important component of Intelligent Transport Systems (ITS) and plays a key role in urban traffic management and provision of traveller information services. Due to its importance, a wide number of researchers have developed different algorithms for real-time incident detection. However, the main limitation with existing techniques is that they do notwork well in conditions where random factors could influence traffic flows. Twitter is a valuable source of information as its users post events as they happen or shortly after. Therefore, Twitter data has been used to predict a wide variety of real-time outcomes. This paper aims to present a methodology for a real-time traffic event detection using Twitter. Tweets are obtained through the Twitter Streaming Application Programming Interface (API) in real-time with a geolocation filter. Then, we used Natural Language Processing (NLP) techniques to process the tweets before they are fed into a text classification algorithm that identifies if its traffic related or not. We implemented our methodology in the West Midlands region in the UK, and obtained an overall accuracy of 92.86%.
Additional Information:	
Question	**Response**
Please enter the total number of words in your main text.	3341
Please enter the number of figures, tables and photographs in your submission.	Tables: 7 Figures: 2

Title: Real-time traffic event detection using Twitter: A case study

Author 1

- Angelica Salas Jones, PhD student
- Faculty of Science and Engineering, University of Wolverhampton, United Kingdom

Author 2

- Dr. Panagiotis Georgakis
- Faculty of Science and Engineering, University of Wolverhampton, United Kingdom

Author 3

- Dr. Ioannis Petalas
- Faculty of Science and Engineering, University of Wolverhampton, United Kingdom

Author 2

- Dr. Renukappa Suresh
- Faculty of Science and Engineering, University of Wolverhampton, United Kingdom

Full contact details of corresponding author.

Email address: a.m.salasjones@wlv.ac.uk

Mobile number: 447481475793

Number of words in main text: 3341

Number of tables: 7

Number of figures: 2

This is an example created from parts of other articles, it is not designed to be read for sense.

Abstract (150 words)

Incident detection is an important component of Intelligent Transport Systems (ITS) and plays a key role in urban traffic management and provision of traveller information services. Due to its importance, a wide number of researchers have developed different algorithms for real-time incident detection. However, the main limitation with existing techniques is that they do notwork well in conditions where random factors could influence traffic flows. Twitter is a valuable source of information as its users post events as they happen or shortly after. Therefore, Twitter data has been used to predict a wide variety of real-time outcomes. This paper aims to present a methodology for a real-time traffic event detection using Twitter. Tweets are obtained through the Twitter Streaming Application Programming Interface (API) in real-time with a geolocation filter. Then, we used Natural Language Processing (NLP) techniques to process the tweets before they are fed into a text classification algorithm that identifies if its traffic related or not. We implemented our methodology in the West Midlands region in the UK, and obtained an overall accuracy of 92.86%.

Keywords

Transport management, Transport planning, Information Technology, Infrastructure planning.

Introduction

With 84% of people travelling by car at least once or twice a week (DfT, 2017), the need for more efficient traffic monitoring systems has become essential. Increases in traffic leads to more interaction between road users, and therefore, heightened likelihood of traffic incidents. Traffic incidents are non-recurrent events such as accidents, broken down vehicles, road maintenance, social activities and other unexpected events that affect the normal traffic flow. These incidents contribute to delays and have serious effects on safety, air pollution and the cost of travel. In order to reduce these adverse effects, incidents need to be detected and cleared as promptly as possible. For these reasons, Automatic Incident Detection (AID) has been widely studied in the last decades. AID is an important part of Intelligent Transportation Systems (ITS), and is designed to automatically detect incidents, or unexpected situations causing congestions in the transport network (D'Andrea, Marcelloni, 2017).

Traditional AID systems exploit data collected from loop detectors and surveillance cameras on the transport network. These devices measure traffic data such as flow, speed, and occupancy for a given period of time. AID algorithms can then detect traffic incidents from anomalies found on these data. However, it is quite expensive to cover broad areas due to the high cost of installing and maintaining these types of devices. In contrast, this approach has poor performance on arterial roads, were traffic flows can be influenced by random factors. Recently, there has been a trend towards considering other data sources technologies, such as GPS and cellular geolocation systems (Parkany, Xie ,2005). Nevertheless, these approaches are limited by low sampling rate and high measurement errors (Siripanpornchana, Panichpapiboon & Chaovalit, 2016).

It would be ideal if users could report incidents in real-time, as they are the ones that can provide more accurate information about the incident. In fact, virtually any person witnessing or involved in any event is able to disseminate it in real-time through microblogs (Atefeh, Khreich ,2015). Microblogging sites, particularly Twitter, have become a popular source of all kinds of information. Twitter is an online social network with over 300 million users posting short messages (tweets) on a real-time basis. Many of these tweets are about real-time events as they happen, or shortly after. For instance, users turn to Twitter to report traffic incidents or to describe the traffic situation they are currently in, making Twitter a real-time source of human travel information. For this reason, Twitter data has proven to be very useful for detecting traffic events. In addition, people use Twitter to express their opinion and emotions on a certain subject. Particularly, traffic related tweets tend to be filled with emotions as users usually complain about the state of the network, or are stressed about a traffic incident. It is important to include this subjective data into traffic incident detection, as it can give a better understanding of the user perception of the transport network (Kokkinogenis et al., 2015).

Using Twitter based data input for traffic incident detection overcome some of the issues faced with conventional devices sensors. First, there is no cost involved as Twitter grants free access to a subset of their data. Second, while traditional sensors only detect changes in traffic measures, a tweet usually contains more detailed information about the traffic event taking place. Third, users can tweet from any location, covering broader areas of the transport network. Lastly, traditional approaches fail to provide an insight into the user's perception of the flaws of the transport network. Nevertheless, there are some challenges involved with using Twitter for incident detection. Traditional text mining techniques do not work well on tweets, as they often contain emoticons, typos, and grammatical errors. Hence, with more than 500 million tweets per day, it is difficult to detect useful information from noise (e.g.: non-traffic related, spams). Finally, although Twitter data is free to access, there is a limitation on the amount that can be obtained in real-time.

This paper presents a methodology for traffic event detection by fetching, filtering and processing public tweets in real-time. The procedure uses Natural Language Processing (NLP) techniques to process the tweets before they are fed into a machine learning classifier. This is an initial attempt to examine the accuracy and potential of incident detection through Twitter. For this reason, although the methodology can be applied in real-time, we implemented it using historical twitter data. The remaining part of the paper proceeds as follows. We first give an overview of different implementations of Twitter for incident detection. The methodology for crawling, processing and classifying tweets is described in section 3. In section 4, results and findings from the experimental implementation are presented. Finally, conclusions and recommendations are drawn.

2. Related work

To date, several studies have analysed the use of Twitter for event detection. (Sakaki, Okazaki & Matsuo, 2010) were amongst the first to propose a methodology to detect events using Twitter. They were able to detect earthquakes with a 96% probability by using a Support Vector Machine (SVM) for classification, and a Kalman filtering for location estimation. (Abel et al. ,2012) developed a framework for filtering, searching, and analysing real-time world incidents from social web streams. Their system could collect Twitter messages, related pictures, and videos to the specific incident. In contrast, (Krstajic et al., 2012) detected potential events by monitoring the frequency of individual keywords and for those with unexpected high frequency values, it calculated additional scores that could help on describing the event. (R. Li et al., 2012) presented TEDAS, a system for detecting, ranking and locating crime and disaster related events by exploring information from Twitter. Similarly, Eventweet focused on detecting events by adopting a continuous analysis of the most recent tweets within a time frame (Abdelhaq, Sengstock & Gert, 2013). Lastly, (Osborne et al., 2014) introduced a system for monitoring security relevant events, and tracking changes in emotions over time.

Concerning traffic incident detection, a number of researchers have presented different methodologies to exploit twitter data as a sensor. For instance, (Gutierrez et al., 2015) described an approach for integrating tweets from different traffic agencies in the UK, with the purpose of notifying drivers about the status of the network in real-time. Our approach concentrates on user generated tweets, rather than official traffic agencies tweets. (Schulz, Ristoski & Paulheim, 2013) presented a methodology for the identification of small scale incidents by combining text classification techniques with a machine learning algorithm. Their outcome was to identify car crashes, while we aim to detect any event that can influence the traffic condition. (D'Andrea et al., 2015) and (Gu, Qian & Chen, 2016) filtered tweets by traffic related keywords, and used a machine learning algorithm to classify them into traffic related or not. (D'Andrea et al., 2015) obtained promising results on the accuracy of the classifier, but they tested it only on the training dataset. In this paper, we test the accuracy of the algorithm on a different dataset, with the purpose of showing that the model is not overfitted to the training data. In addition, these studies used the Twitter REST API to crawl tweets, while we propose fetching them through the Twitter Streaming API. Lastly, existing research for mining user generated tweets for traffic incident has been applied in the United States, Italy, and Germany. In this study, we employed the methodology in the West Midlands region, in the United Kingdom.

3. Methodology

In this section, we describe the methodology used to identify traffic incident information from twitter data. Figure 1 shows the system architecture and the different tools used on each phase. We fetched tweets using the Twitter Streaming API with a geolocation filter. Road names and traffic related words were used as keywords as an additional filter. Next, we trained five machine learning algorithms with different word n-grams and tested their classification accuracy. Finally, we selected the most accurate n-gram features, and evaluated each classifier on the test dataset.

3.1 Fetching Twitter data

The first step entails the extraction of raw tweets using the Twitter Streaming API. One of the limitations of using the Streaming API is that it does not allow to filter by location and keyword. This is the main reason why authors in the literature have used the Twitter Search API for their studies. However, the Search API searches against a sample of recent tweets focusing on their relevance, while the streaming API gives real-time access to the streams of public data flowing through Twitter (Twitter, 2017). For this reason, we selected the Streaming API for this stage. Twitter API’s are supported in many programming languages through a wide variety of libraries. In our approach, we made an uninterrupted connection to the Streaming API with a geolocation filter, using the Tweepy library in Python.

3.2 Traffic keywords filtering

On this stage, we perform an additional filter to obtain the tweets mentioning traffic related words. To this end, we created a dictionary of highways, arterials, roads, and incident related words. We used regular expressions to filter the acquired tweets using the keyword dictionary. Table 1 shows an example of some of the keywords and road names used for filtration. In addition, we used this stage to remove all retweets, as they do only contain repeated information.

Traffic keywords	Road names
Accident	M6
Congestion	A449
Roadworks	M42
Traffic delays	A41
Stuck traffic	M5

Table 1: Keywords for filtration

3.3 Pre-processing

Due to their informal nature, tweets usually contain mentions, hashtags, links, special characters and emoticons. This information needs to be removed before tweets are fed into the classifier. In the following sections, the text mining techniques applied to the dataset are described in detail.

3.3.1 Tokenisation

Tokenisation is the task of transforming a character sequence into pieces, called tokens, and at the same time removing certain characters. There are a wide range of tokenization tools, however they fail to recognise special tweet features such as @mentions, emoticons, URLs and hashtags as individual tokens. For this reason, we employed a pre-processing chain based on regular expressions that considers all these aspects. During this step, the tokeniser removes mentions, hashtags, URLs, punctuation and emoticons, and splits each tweet into a set of words ('tokens').

3.3.2 Stop word removal

Stop words are those common words that have little value in helping characterise a text, such as articles, conjunctions, and prepositions. These words are not very meaningful when deciding if a tweet is traffic related or not, thus not valuable to be fed into a machine learning classifier. In our approach, the full list of English stop words from the Natural Language Toolkit (NLTK) library was used to remove stop words from the set of tokens.

3.4 Classification

Once tweets have been pre-processed, they were classified into traffic related or not. To achieve this, a machine learning algorithm was employed. Studies in the literature have employed and compared a wide range of text classification algorithms for incident detection using Twitter data (Schulz, Ristoski & Paulheim, 2013, D'Andrea et al., 2015, Wanichayapong et al., 2011, Gu, Qian & Chen, 2016). For this study, we compared a Ridge Classifier (RC), Naïve Bayes (NB), k-Nearest Neighbour (kNN), Multilayer Perceptron (MLP) and a Support Vector Machine (SVM). We combined and evaluated the classifiers with different word n-gram features on the training dataset, and selected the most accurate parameters on each algorithm for the test data. For this step, the machine learning library ScikitLearn was used.

4. Case study: West midlands region, England

We evaluated our methodology using tweets from the West Midlands area in the United Kingdom. Firstly, we measured the performance of the classifiers using different features on the training dataset. Then, we selected the most effective feature amongst each classifier for the test dataset. Lastly, we compared our work to similar studies in the literature.

4.1 Twitter data acquisition

We collected 4 million tweets using an uninterrupted connection to the Twitter Streaming API from March 1st, 2017 to May 31st, 2017, with the coordinates to the West Midlands region as a geolocation filter. From these data, the regular expressions filter extracted 13,410 tweets, using a dictionary of 265 road names and traffic related keywords. Tweets were then manually labelled into traffic and non-traffic related, and divided into the following datasets:

- Training: This is the portion of tweets used to train and validate the text classification algorithms. It consisted of 785 traffic related tweets and 785 non-traffic related.

- Test: To show the effectiveness of the classifiers on a different dataset than the training, we built a test dataset of 196 traffic tweets and 196 non-traffic related tweets.

From the three-month period, May obtained the highest amount of traffic related Tweets (see figure 1). This was influenced by a high traffic of tweets on the 15th and 16th of May, due to the identification of an undetonated WWII bomb in the city centre of Birmingham. Table 2 has some examples of traffic and non-traffic related tweets from May 2017. It is important to mention that even though our methodology does not include geolocation, we only took into consideration as traffic tweets those that specify the location of the incident.

Tweet	Label
Brum traffic chaos all entry and exit slip roads to m6 at spaghetti junction and the whole a38m are closed due to a bomb being found #ww2	Traffic
massive car crash on pedmore road by merry hill going towards halesowen road all shut off so avoid it	Traffic
traffic chaos bingo big delays in #birmingham #ww2bomb #aston	Traffic
just heard… interview car crash is an understatement	Non-traffic
after a few rough days following my crash im working hard staying positive to get fixed for	Non-Traffic

Table 2: Examples of tweets and their label

4.2 Experimental results

With the purpose of identifying which feature works best with the different machine learning algorithms, we tested each classifier with different n-gram values on the training dataset. For this step, we used a k-fold cross validation methodology. K-fold crossvalidation randomly partitions the dataset into k equal sized folds. From these folds, one is retained for testing the model, while the remaining k-1 are used as training data. This process is repeated k times, using each of the k folds exactly once as test data. We performed the k-fold crossvalidation with n = 10 on the training dataset for each classifier/n-gram.

To evaluate the performance of the classifiers, we calculated the statistical metrics shown in table 3. True negative (TN) and true positive (TP) correspond to the tweets that were classified correctly as non-traffic and traffic related, respectively; while False negative (FN) and False positive (FP) tweets are those that were misclassified as non-traffic and traffic tweets. Accuracy is the overall efficiency of the classifier and corresponds to the fraction of correctly classified tweets by the total number of tweets. Precision of a class represents the fraction of correctly classified tweets within that class. Recall of a class is the number of correctly classified tweets over the total number of tweets that belong to that class. F1-score is the weighted mean of precision and recall.

Metric	**Formula**
Accuracy	$acc = \frac{(TP+TN)}{(TP+FP+TN+FN)}$
Precision	$Prec = \frac{TP}{TP + FP}$
Recall	$Rec = \frac{TP}{TP + FN}$
F1 score	$F1 = \frac{2 \times P \times R}{P + R}$

Table 3: Evaluation metrics

Table 4 shows the results from the cross validation of the training data using different n-gram ranges. For each classifier, we performed the 10-fold cross validation using unigrams, bigrams, unigrams and bigrams, and unigrams, bigrams and trigrams. We calculated the average of the 10 values of accuracy obtained in the cross validation. It can be perceived that most of the classifiers have higher perfomance using unigrams or the combination of the three features, while the worst performance amongst all is observed on the trigrams.

Model	Unigrams	Bigrams	Trigrams	Unigrams and Bigrams	Unigrams, Bigrams and Trigrams
RC	**90.19%**	84.92%	64.20%	90.04%	88.87%
KNN	86.88%	77.78%	50.38%	87.83%	**87.90%**
NB	87.96%	78.30%	56.18%	88.66%	**88.98%**
MLP	89.49%	84.87%	64.59%	**90.32%**	89.87%
SVM	**90.32%**	84.82%	64.20%	89.77%	88.64%

Table 4: Classifiers vs word n-gram features

We selected the feature with the highest accuracy for each classifier, and proceeded to evaluate them on the test dataset. Table 5 depicts the classification results for each classifier on the test dataset. The classifier with the highest accuracy was the Ridge classifier (RC) with a 92.86%. MLP and SVM had similar performance to the Ridge classifier both with 92.6%, while the NB was the one with the lowest accuracy with an 89.54%. These results show that the classifiers are not overfitted to the events in the training data. The classifiers had more precision predicting non-traffic related tweets, but less recall. This shows that while the model identified a higher number of traffic related tweets, they had more precision identifying non-traffic related ones.

Model	Traffic			Non-Traffic			Accuracy
	Prec	Rec	F1	Prec	Rec	F1	
RC	90.38%	95.92%	93.07%	95.65%	89.80%	92.63%	**92.86%**
KNN	86.18%	95.41%	90.56%	94.86%	84.69%	89.49%	90.05%
NB	84.14%	97.45%	90.31%	96.97%	81.63%	88.64%	89.54%
MLP	89.57%	96.43%	92.87%	96.13%	88.78%	92.31%	**92.60%**
SVM	89.57%	96.43%	92.87%	96.13%	88.78%	92.31%	**92.60%**

Table 5: Results on the test dataset

Results from the test dataset showed that a RC, MLP or a SVM would obtain high accuracy on classifying tweets into traffic related or not. However, there are other aspects that need to be taken into consideration, such as the training and prediction time. Table 6 contains the training and prediction time of each algorithm on the test dataset in seconds. RC and SVM are the fastest in both training and prediction both with 0.04s and 0.008s respectively. However, although MLP obtained one of the highest accuracy scores, it needed 43.53s to train. This is more than 1000 times more of what was needed by the RC and the SVM. Contrary to RC and

SVM, MLP obtained more accuracy using unigrams and bigrams, instead of only unigrams, which increases the computing time. However, MLP always obtained the highest computing time amongst all the n-gram variations.

Algorithm	Training time	Prediction time
RC	**0.044**	**0.008**
KNN	0.149	0.048
NB	0.176	0.039
MLP	43.53	0.02
SVM	**0.04**	**0.008**

Table 6: Training and prediction time (sec)

As seen in table 7, results from our RC outperformed studies in the literature. We only took into consideration studies that tested their classifiers on a dataset different than the training one. (Gu, Qian & Chen ,2016) obtained an accuracy of 90.5% on their test dataset, using a Naïve Bayes classifier identifying traffic related tweets. On the other hand, (Schulz, Ristoski & Paulheim ,2013) compared SVM, RIPPER and NB for the identification of car accidents, with accuracies of 89.06%, 84.21% and 79.21%, respectively. In this paper, we used a split of 75%/25% of the train and test data, which was similar to the ones used by these studies. Both studies employed the REST API for crawling tweets, while we used the Streaming API.

Author	Algorithm	Train/Test split	Accuracy
(Gu, Qian & Chen ,2016)	Naives Bayes	77.5%/22.5%	90.5%
(Schulz, Ristoski & Paulheim, 2013)	Support Vector Machine	75%/25%	89.06%
	RIPPER		84.21%
	Naïves Bayes		79.21%

Table 7: Results from the literature

4. Conclusions and future work

We have developed a methodology for crawling, processing, and classifying traffic related tweets in real-time. We fetched tweets using an uninterrupted connection to the Streaming API. Then, we used natural language processing techniques to remove special characters and stop-words. We compared five different machine learning algorithms, and obtained an overall highest accuracy of 92.86% with a Ridge Classifier on our test data. Our results outperformed similar studies in the literature.

Our experimental results show the ability of the system in detecting traffic incidents on real-time. This information can be incorporated on AID systems to improve their accuracy to wider areas of the network. Social media data can also be used to detect the feedback of the users in specific parts of the network.

This paper is part of an on-going work for a real-time pipeline for incident detection using Twitter. Future work includes the use of additional NLP techniques to improve the accuracy of the classifier and to detect the location of the incident. Finally, sentiment and stress analysis will be performed to obtain the user's perspective of the network.

Acknowledgements

This research was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 636160-2, the Optimum project www.optimumproject.eu.

This paper is part of a PhD sponsored by the Dominican Republic's Ministry of Education (MESCyT).

References

Abdelhaq, H., Sengstock, C. & Gertz, M. 2013, "Eventweet: Online localized event detection from twitter", *Proceedings of the VLDB Endowment,* vol. 6, no. 12, pp. 1326-1329.

Abel, F., Hauff, C., Houben, G., Stronkman, R. & Tao, K. 2012, "Twitcident: fighting fire with information from social web streams", *Proceedings of the 21st International Conference on World Wide Web*ACM, , pp. 305.

Atefeh, F. & Khreich, W. 2015, "A Survey of Techniques for Event Detection in Twitter", *Computational Intelligence,* vol. 31, no. 1, pp. 132-164.

D'Andrea, E., Ducange, P., Lazzerini, B. & Marcelloni, F. 2015, "Real-Time Detection of Traffic From Twitter Stream Analysis", *IEEE Transactions on Intelligent Transportation Systems,* vol. 16, no. 4, pp. 2269-2283.

D'Andrea, E. & Marcelloni, F. 2017, "Detection of traffic congestion and incidents from GPS trace analysis", *Expert Systems with Applications,* vol. 73, pp. 43-56.

DfT 2017, *Road Traffic Estimates: Great Britain 2016*. Available: https://www.gov.uk/government/statistics/road-traffic-estimates-in-great-britain-2016

Gu, Y., Qian, Z.(. & Chen, F. 2016, "From Twitter to detector: Real-time traffic incident detection using social media data", *Transportation Research Part C: Emerging Technologies,* vol. 67, pp. 321 342.

Gutierrez, C., Figuerias, P., Oliveira, P., Costa, R. & Jardim-Goncalves, R. 2015, "Twitter mining for traffic events detection", *Proceedings of the 2015 Science and Information Conference, SAI 2015,* , pp. 371-378.

Krstajic, M., Rohrdantz, C., Hund, M. & Weiler, A. 2012, "Getting there first: Real-time detection of real-world incidents on twitter", .

N. Wanichayapong, W. Pruthipunyaskul, W. Pattara-Atikom & P. Chaovalit 2011, "Social-based traffic information extraction and classification", *ITS Telecommunications (ITST), 2011 11th International Conference on*, pp. 107.

Osborne, M., Moran, S., McCreadie, R., Von Lunen, A., Sykora, M.D., Cano, E., Ireson, N., Macdonald, C., Ounis, I. & He, Y. 2014, "Real-time detection, tracking, and monitoring of automatically discovered events in social media", .

Parkany, E. & Xie, C. 2005, *A complete review of incident detection algorithms & their deployment: what works and what doesn't*.

R. Li, K. H. Lei, R. Khadiwala & K. C. C. Chang 2012, "TEDAS: A Twitter-based Event Detection and Analysis System", *2012 IEEE 28th International Conference on Data Engineering*, pp. 1273.

Sakaki, T., Okazaki, M. & Matsuo, Y. 2010, "Earthquake shakes Twitter users: real-time event detection by social sensors", *Proceedings of the 19th international conference on World wide web*ACM, , pp. 851.

Schulz, A., Ristoski, P. & Paulheim, H. 2013, "I See a Car Crash: Real-Time Detection of Small Scale Incidents in Microblogs", Springer, Berlin, Heidelberg, , pp. 22.

Siripanpornchana, C., Panichpapiboon, S. & Chaovalit, P. 2016, "Incidents detection through mobile sensing", IEEE, , pp. 1.

Twitter 2017. *The Search API*. Available: https://dev.twitter.com/rest/public/search.

Figure 1. Tweets per month

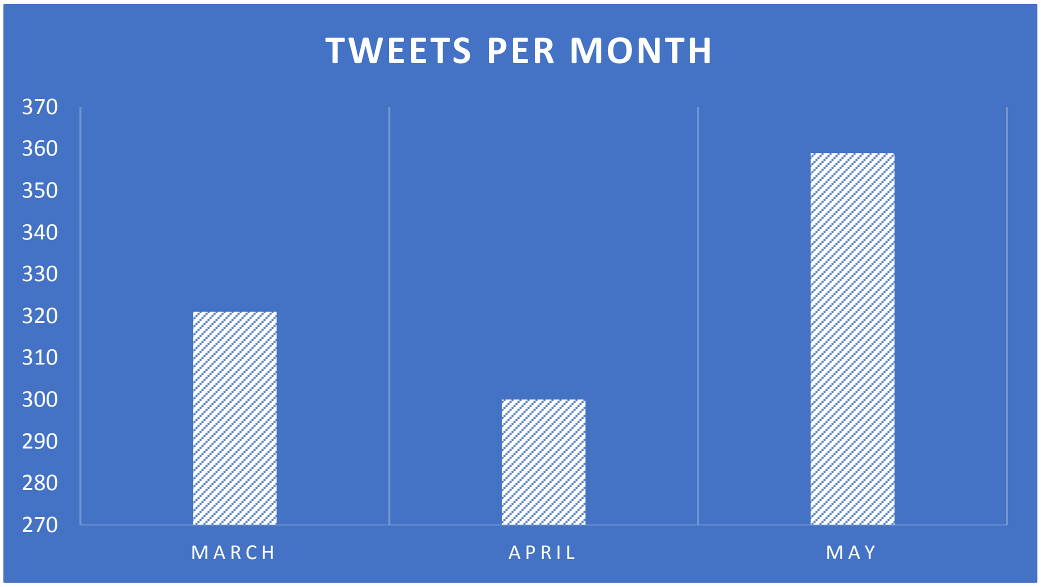

Figure 2. System architecture